REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-71 are pending in the application. The Examiner additionally stated that claims 1-71 are rejected. By this amendment, claims 50-51 have been cancelled and claims 53 and 71 have been amended. Hence, claims 1-49 and 52-71 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Claims

Rejections Under 35 U.S.C. §112 second paragraph and 35 U.S.C. §101

The Examiner rejected claims 50-51 under 35 U.S.C. 112 second paragraph as being indefinite and under 35 U.S.C. 101 as not being limited to tangible embodiments. Applicant has canceled claims 50-51.

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-8, 10, 13-27, 29-34, 39-52, and 54-70 under 35 U.S.C. 102(b) as being anticipated by Hubis et al, U.S. Patent No. 6,343,324 (hereinafter, *Hubis*). Applicant respectfully traverses the Examiner's rejections.

Hubis is directed to a controller 106, such as a disk array controller, that includes a processor 180 and memory 181 and 182 for storing a computer program to be executed by the processor 180 that controls host I/O processors 184 (e.g., Fibre Channel or SCSI I/O processors) that interface the controller 106 to host computers, and that controls storage I/O processors 185 (e.g., Fibre Channel or SCSI I/O processors) that interface the controller 106 to storage devices. Fig. 2A; col. 15, line 53 to col. 16, line 9; col. 11, lines 22-30.

The controller 106 of *Hubis* enables multiple host computers to share access to the storage devices attached to the controller 106. The controller 106 controls access by the hosts to the storage devices based on unique host identifiers identifying each of the hosts so that information on the storage devices will not be corrupted by unauthorized hosts

accessing the storage devices. When the controller 106 receives a command from a host specifying the SCSI LUN to access, the controller 106 performs a host-to-volume mapping function to determine which logical volume on the storage devices the command is directed towards. Based on the mapping, the controller 106 determines whether the particular host has permission to access the logical volume. Fig. 3B, steps 317 to 326; Fig. 2B-3; Abstract; Col. 4, lines 40-54; col. 3, lines 37-47; col. 5, lines 12-18, 48-52; col. 6, lines 9-14; col. 7, lines 22-37; col. 9, line 65 to col. 10, line 2; col. 11, lines 37-57.

The processor 180 of *Hubis*' controller 106 executes a computer program (also referred to in *Hubis* as "firmware", a specific example of which is the MYLEX DAC RAID controller firmware version 5.0) that performs the host-to-volume mapping and access control functions in addition to handling the normal storage controller tasks including data reads and writes, error handling, cache memory management, RAID control, and controlling the Fibre Channel or SCSI I/O processors 184 and 185 that transfer commands and data with the host computers and the physical storage devices. Col. 15, line 53 to col. 16, line 9; col. 5, lines 22-42; col. 11, lines 59-64; col. 14, lines 13-37; col. 7, lines 37-38; col. 4, line 64 to col. 5, line 3. Thus, the controller 106 taught by *Hubis* performs the access control and mapping functions in the same manner described in Applicants' Background section in paragraphs [0007] to [0008], namely using the same microprocessor of the controller 106 that performs other storage controller functions, such as RAID functions, cache buffer management, and data transfers between the storage devices and the controller and the hosts and the controller. Consequently, the controller 106 of *Hubis* may suffer from the same problems described in Applicants' Background section in paragraph [0009], namely that the microprocessor may become the performance bottleneck.

Advantageously, Applicants teach and claim a way to offload the microprocessor from performing the access control and mapping functions, namely by having the host interface adapter perform those functions. Specifically, *Hubis* does not teach that his host I/O processors 184 perform access control or mapping functions.

With respect to claim 1, the Examiner states that *Hubis* teaches a host interface adapter that determines whether a host computer identified in a data transfer request received from the host computer is allowed to access a logical storage device identified in the request, citing col. 11, lines 45-53 of *Hubis*. Applicant respectfully asserts that *Hubis* does not teach a host interface adapter that determines whether a host computer identified in a data transfer request received from the host computer is allowed to access a logical storage device identified in the request, neither at col. 11, lines 45-53 nor elsewhere.

Hubis teaches at col. 11, lines 45-53 that the "controller 106" determines whether the host computer is allowed to access the logical volume, and does not specify which element of the controller 106 performs this function. However, and as discussed above, *Hubis* teaches elsewhere that it is his processor 180 that decides whether to allow a host access to a logical volume. Col. 15, line 53 to col. 16, line 9; col. 5, lines 22-42; col. 11, lines 59-64; col. 14, lines 13-37. As discussed above, overburdening the microprocessor with the access control function is precisely the problem Applicants' invention is solving, see paragraph [0009], by offloading this function to the host interface adapter. *Hubis*' host interface adapters are host I/O processors 184, and Applicant can find no teaching in *Hubis* that the host I/O processors 184 decide whether to allow a host access to a logical storage device.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 1.

With respect to claim 4, Applicant respectfully asserts that *Hubis* does not teach a host interface adapter that provides an indication that a host computer identified in a request is not allowed to access a logical storage device identified in the request.

With respect to claim 18, for reasons similar to those discussed below with respect to claim 67, Applicant respectfully asserts that *Hubis*[1] does not teach a storage controller comprising a plurality of microprocessors, nor does *Hubis* teach a host interface adapter that determines which one of the plurality of microprocessors is configured to process requests for the logical storage device identified in the request.

[1] Applicant assumes the Examiner intended to recite *Hubis* rather than *Kitamura* with respect to claim 18.

With respect to claim 46, for reasons similar to those discussed below with respect to claim 54, Applicant respectfully asserts that *Hubis* does not teach an interface adapter that maps a first identifier to a second identifier, wherein the first identifier is included in a host request and is used by the host to specify one of the logical storage devices, wherein the second identifier is used by the storage controller to specify one of the logical storage devices.

With respect to claim 48, Applicant respectfully asserts that *Hubis* does not teach an interface adapter comprising a memory for storing an access table specifying which of the hosts has access to which of the logical storage devices.

Claims 2-23 depend from independent claim 1, which is not anticipated by *Hubis* as discussed above, and add further limitations. Therefore, Applicant respectfully asserts that claims 2-23 are not anticipated by *Hubis*, and respectfully requests the Examiner withdraw the rejections.

With respect to claim 24, the Examiner states that *Hubis* teaches an interface adapter that determines from an access table whether a host requesting access has access to a logical storage device, citing col. 3, lines 43-58 of *Hubis*. Applicant respectfully asserts that *Hubis* does not teach an interface adapter that determines from an access table whether a host requesting access has access to a logical storage device, neither at col. 3, lines 43-58 nor elsewhere.

Hubis teaches at col. 3, lines 43-58 in his Background section the problem solved by *Hubis*' invention, namely data corruption caused by multiple hosts sharing access to logical volumes. Furthermore, *Hubis* teaches elsewhere that it is his processor 180 that determines from various tables whether a host has access to a logical volume. Col. 15, line 53 to col. 16, line 9; col. 5, lines 22-42; col. 11, lines 59-64; col. 14, lines 13-37. *Hubis*' host interface adapters are host I/O processors 184, and Applicant can find no teaching in *Hubis* that the host I/O processors 184 determine from an access table whether a host requesting access has access to a logical storage device.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 24.

Claims 25-49 depend from independent claim 1, which is not anticipated by *Hubis* as discussed above, and add further limitations. Therefore, Applicant respectfully asserts that claims 25-49 are not anticipated by *Hubis*, and respectfully requests the Examiner withdraw the rejections.

With respect to claim 52, the Examiner states that *Hubis* teaches an interface adapter determining whether a host computer requesting access has access to a logical storage device, citing col. 11, lines 45-53 and col. 9, lines 52-57 of *Hubis*. For the reasons discussed above with respect to claim 1, Applicant respectfully asserts that *Hubis* does not teach an interface adapter determining whether a host computer requesting access has access to a logical storage device, neither at col. 11, lines 45-53 nor elsewhere. Furthermore, at col. 9, lines 52-57 *Hubis* simply teaches that his controller 106 performs host access control, but does not teach an interface adapter performing the access control.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 52.

With respect to claim 54, the Examiner states that *Hubis* teaches a host interface adapter that takes an external identifier included in a host request that identifies a storage device the host is requesting to access and maps the external identifier to an internal identifier by which the microprocessor of the storage controller uniquely identifies the storage device, citing col. 11, lines 37-53 of *Hubis*. Applicant respectfully asserts that *Hubis* does not teach a host interface adapter that takes an external identifier included in a host request that identifies a storage device the host is requesting to access and maps the external identifier to an internal identifier by which the microprocessor of the storage controller uniquely identifies the storage device, neither at col. 11, lines 37-53 nor elsewhere.

Hubis teaches at col. 11, lines 37-42 that: "Controller 106 uses the LUN number requested by the host and the identity of the host-to-controller port 114, 115 (or 184, 185) at which the command was received, both of which are produced by the ISP with [the] command, to determine which Logical Volume the host is trying to access." The "ISP" is host I/O processor 184. See col. 11, lines 30-37. First, it is apparent from the cited text that the host I/O processor 184 does not map the LUN number and port 114 identity to

map to the logical volume to be accessed, because the text states that the host I/O processor 184 "provides" the LUN number and port 114 identity to something else that performs the mapping function. Second, as discussed above, *Hubis* explicitly teaches that it is his processor 180, rather than the host I/O processor 184, that maps the LUN and port 114 identifier to the logical volume. Col. 15, lines 45-63; col. 5, lines 22-27; col. 14, lines 13-37. As discussed above, overburdening the microprocessor with the mapping function is precisely the problem Applicants' invention is solving, paragraph [0009] by offloading this function to the host interface adapter. *Hubis*' host interface adapters are host I/O processors 184, and Applicant can find no teaching in *Hubis* that the host I/O processors 184 map the external identifier to an internal identifier by which the microprocessor of the storage controller uniquely identifies the storage device.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 54.

Claims 55-65 depend from independent claim 54, which is not anticipated by *Hubis* as discussed above, and add further limitations. Therefore, Applicant respectfully asserts that claims 55-65 are not anticipated by *Hubis*, and respectfully requests the Examiner withdraw the rejections.

With respect to claim 66, the Examiner states that *Hubis* teaches a host interface adapter mapping a combination of a host computer identifier and a storage device identifier to a unique identifier used by the microprocessor of the storage controller to identify the storage device, citing col. 11, lines 8-15 and 38-44 of *Hubis*. Applicant respectfully asserts that *Hubis* does not teach a host interface adapter mapping a combination of a host computer identifier and a storage device identifier to a unique identifier used by the microprocessor of the storage controller to identify the storage device, neither at col. 11, lines 8-15 and 38-44 nor elsewhere.

As discussed above with respect to claim 54, it is apparent from *Hubis* teaching at col. 11, lines 37-42 that the host I/O processor 184 does not map the LUN number and port 114 identity to map to the logical volume to be accessed, because the text states that the host I/O processor 184 "provides" the LUN number and port 114 identity to something

else that performs the mapping function. And, *Hubis* explicitly teaches that it is his processor 180, rather than the host I/O processor 184, that maps the LUN and port 114 identifier to the logical volume. Col. 15, lines 45-63; col. 5, lines 22-27; col. 14, lines 13-37. Finally, at col. 11, lines 8-15 *Hubis* merely states that the mapping table and mapping table entries are instantiated for the controller 106, but does not state that the host I/O processor 184 uses the mapping table to map anything. Rather, *Hubis* teaches that the processor 180 uses the mapping table entries to map the LUN number and port 114 identity to map to the logical volume. Col. 15, lines 45-63; col. 5, lines 22-27; col. 14, lines 13-37. Applicant can find no teaching in *Hubis* of a host interface adapter mapping a combination of a host computer identifier and a storage device identifier to a unique identifier used by the microprocessor of the storage controller to identify the storage device.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 66.

With respect to claim 67, the Examiner states that *Hubis* teaches a storage controller comprising first and second microprocessors, citing col. 11, lines 8-15, and stating that *Hubis* teaches multiple controllers thereby anticipating the instant limitation. Applicant respectfully asserts that *Hubis* does not teach a storage controller comprising first and second microprocessors, neither at col. 11, lines 8-15 nor elsewhere.

Hubis does teach multiple controllers 106 each having a single microprocessor 180. However, *Hubis* does not teach a single controller 106 having more than one microprocessor 180; whereas, claim 67 recites a storage controller having first and second microprocessors. That is, claim 67 does not recite multiple storage controllers having first and second microprocessors. Therefore, Applicant respectfully asserts the Examiner has improperly stated that *Hubis* anticipates the limitation.

Further with respect to claim 67, the Examiner states that *Hubis* teaches an interface adapter that determines whether a request specifies a logical storage device in a first or second set of logical storage devices for whom requests are respectively processed by the first and second microprocessors, citing col. 11, lines 8-15 and 45-53 of *Hubis*.

Applicant respectfully asserts that *Hubis* does not teach an interface adapter that determines whether a request specifies a logical storage device in a first or second set of logical storage devices for whom requests are respectively processed by the first and second microprocessors, neither at col. 11, lines 8-15 and 45-53 nor elsewhere. This follows from the fact that *Hubis* does not teach a single storage controller with multiple microprocessors, as asserted above.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 67.

Claims 68-70 depend from independent claim 67, which is not anticipated by *Hubis* as discussed above, and add further limitations. Therefore, Applicant respectfully asserts that claims 68-70 are not anticipated by *Hubis*, and respectfully requests the Examiner withdraw the rejections.

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claims 9 and 28 under 35 U.S.C. 103(a) as being unpatentable over *Hubis* in view of Camble et al, U.S., Patent No. 6,999,999 (hereinafter *Camble*). Applicant respectfully traverses the Examiner's rejections.

The Examiner rejected claims 11-12, 35-38, 53, and 71 under 35 U.S.C. 103(a) as being unpatentable over *Hubis* in view of Kitamura et al, U.S., Publication No. 2002/0199071 (hereinafter *Kitamura*). Applicant respectfully traverses the Examiner's rejections.

With respect to claim 53, Applicant has amended the claim to more distinctly claim the subject matter Applicant regards as his invention. For reasons similar to those stated above with respect to claim 1, Applicant respectfully asserts that *Hubis* in view of *Kitamura* does not obviate claim 53. In particular, *Hubis* does not teach an interface adapter operable to implement host access controls for storage space on storage devices. For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 53.

With respect to claim 71, Applicant has amended the claim to more distinctly claim the subject matter Applicant regards as his invention. For reasons similar to those stated

above with respect to claim 54, Applicant respectfully asserts that *Hubis* in view of *Kitamura* does not obviate claim 71. In particular, *Hubis* does not teach an interface adapter operable to map between hosts and storage devices. For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 71.

CONCLUSIONS

In view of the arguments advance above, Applicant respectfully submits that claims 1-49 and 52-71 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

Respectfully submitted,

/E. Alan Davis/

By:_____

E. ALAN DAVIS
Registration No. 39,954
Tel: (512) 301-7234

10/3/2006

Date: _____